A FAST FREQUENCY TRANSFORMATION TECHNIQUE FOR TRANSFORM AUDIO CODERS

Technical Field

This invention is applicable in the field of multi-channel audio coders which use modified discrete cosine transform as a step in the compression of audio signals.

Background Art

In order to more efficiently broadcast or record audio signals, the amount of information required to represent the audio signals may be reduced. In the case of digital audio signals, the amount of digital information needed to accurately reproduce the original pulse code modulation (PCM) samples may be reduced by applying a digital compression algorithm, resulting in a digitally compressed representation of the original signal. The goal of the digital compression algorithm is to produce a digital representation of an audio signal which, when decoded and reproduced, sounds the same as the original signal, while using a minimum of digital information for the compressed or encoded representation.

Recent advances in audio coding technology have led to high compression ratios while keeping audible degradation in the compressed signal to a minimum. These coders are intended for a variety of applications, including 5.1 channel film soundtracks, HDTV, laser discs and multimedia. Description of one applicable method can be found in the Advanced Television Systems Committee (ATSC) Standard document entitled "Digital Audio Compression (AC-3) Standard", Document A/52, 20 December, 1995.

In the basic approach, at the encoder the time domain audio signal is first converted to the frequency domain using a bank of filters. The frequency domain coefficients, thus generated, are converted to fixed point representation. In fixed point syntax, each coefficient is represented as a mantissa and an exponent. The bulk of the compressed bitstream transmitted to the decoder comprises these exponents and mantissas.

The exponents are usually transmitted in their original form. However, each mantissa must be truncated to a fixed or variable number of decimal places. The number of bits to be used for coding each mantissa is obtained from a bit allocation algorithm which may be based on the masking property of the human auditory system. Lower numbers of bits result in higher compression ratios because less space is required to transmit the coefficients. However, this may cause high quantization errors, leading to audible distortion. A good distribution of available bits to each mantissa forms the core of the advanced audio coders.

The frequency transformation phase has one of the greatest computation requirements in a transform coder. Therefore, an efficient implementation of this phase can decrease the computation requirement of the system significantly and make real time operation of the encoder more easily attainable.

In some encoders such as those specified in the AC-3 standard, the frequency domain transformation of signals is performed by the modified discrete cosine transform (MDCT). If directly implemented, the MDCT requires $O(N^2)$ additions and multiplications. However it has been found possible to reduce the number of required operations significantly if the MDCT equation is able to be computed in a from that is amenable to the use of the well known Fast Fourier Transform (FFT) method of J.W. Cooley and J.W. Tukey (1960). A known application of an FFT to an MDCT is disclosed in, for example, EP-A-0564089.

The present invention seeks to provide an alternative computation method using a Fast Fourier Transform. Moreover, the invention seeks to use a single FFT for two channels to achieve greater reduction in computational requirements of the system.

Summary of the Invention

In accordance with the present invention there is provided a method for coding audio data comprising a sequence of digital audio samples, including the steps of:

i) multiplying the input samples with a first trigonometric function factor to generate an intermediate sample sequence;

ii) computing a fast Fourier transform of the intermediate sample sequence to generate a Fourier transform coefficient sequence;

iii) for each transform coefficient in the sequence, multiplying the real and imaginary components of the transform coefficient by respective second trigonometric function factors, adding the multiplied real and imaginary transform coefficient components to generate an addition stream coefficient, and subtracting the multiplied real and imaginary transform coefficient components to generate a subtraction stream coefficient;

iv) multiplying the addition and subtraction stream coefficients with respective third trigonometric function factors; and

v) subtracting the corresponding multiplied addition and subtraction stream coefficients to generate audio coded frequency domain coefficients.

The present invention also provides a method for coding audio data, including the steps of:

combining first and second sequences of digital audio samples from first and second audio channels into a single complex sample sequence;

determining a Fourier transform coefficient sequence as defined above;

generating first and second transform coefficient sequences by combining and/or differencing first and second selected transform coefficients from said Fourier transform coefficient sequence; and

for each of the first and second transform coefficient sequences, generating audio coded frequency domain coefficients as defined above, so as to generate respective sequences of said audio coded frequency domain coefficients for the first and second audio channels.

The present invention also provides a method for coding audio data including the steps of:

obtaining at least one input sequence of digital audio samples;

pre-processing the input sequence samples including applying a pre-multiplication factor to obtain modified input sequence samples;

transforming the modified input sequence samples into a transform coefficient sequence utilising a fast Fourier transform; and

post-processing the sequence of transform coefficients including applying first post-multiplication factors to the real and imaginary coefficient components, differencing and combining the post-multiplied real and imaginary components, applying second post-multiplication factors to the difference and combination results, and differencing to obtain a sequence of modified discrete cosine transform coefficients representing said input sequence of digital audio samples.

The present invention also provides a method for coding audio data including the steps of:

obtaining first and second input sequences of digital audio samples corresponding to respective first and second audio channels;

combining the first and second input sequences of digital audio samples into a single complex input sample sequence;

pre-processing the complex input sequence samples including applying a pre-multiplication factor to obtain modified complex input sequence samples;

transforming the modified complex input sequence samples into a complex transform coefficient sequence utilising a fast Fourier transform; and

post-processing the sequence of complex transform coefficients to obtain first and second sequences of audio coded frequency domain coefficients corresponding to the first and second audio channels including, for each corresponding frequency domain coefficient in the first and second sequences, selecting first and second complex transform coefficients from said sequence of complex transform coefficients, combining the first complex transform coefficient and the complex conjugate of the second complex transform coefficient for said first channel and differencing the first complex transform coefficient and the complex conjugate of the second complex transform coefficient for said second channel, and applying respective post-multiplication factors to the combination and difference to obtain said audio coded frequency domain coefficients corresponding to the first and second audio channels.

The present invention further provides A method for coding audio data including the steps of:

obtaining first and second input sequences of digital audio samples $x[n]$, $y[n]$ corresponding to respective first and second audio channels;

combining the first and second input sequences of digital audio samples into a single complex input sample sequence $z[n]$, where $z[n] = x[n] + jy[n]$;

pre-processing the complex input sequence samples including applying a pre-multiplication factor $cos(\pi n/N) + jsin(\pi n/N)$ to obtain modified complex input sequence samples, where N is the number of audio samples in each of the first and second input sequences and $n = 0,\ldots,(N\text{-}1)$;

transforming the modified complex input sequence samples into a complex transform coefficient sequence Z_k utilising a fast Fourier transform, wherein $k = 0,\ldots,(N/2\text{-}1)$; and

post-processing the sequence of complex transform coefficients to obtain first and second sequences of audio coded frequency domain coefficients corresponding to the first and second audio channels X_k, Y_k according to:

$$G_k = (Z_k + Z^{*}_{N-k-1})/2 \qquad k=0..N/2-1$$

$$G'_k = (Z_k - Z^{*}_{N-k-1})/2j \qquad k=0..N/2-1$$

$$\begin{aligned} X_k = {} & \cos\gamma * (g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N) \\ & - \sin\gamma * (g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N) \end{aligned}$$

$$\begin{aligned} Y_k = {} & \cos\gamma * (g'_{k,r}\cos(\pi(k+1/2)/N) - g'_{k,i}\sin(\pi(k+1/2)/N) \\ & - \sin\gamma * (g'_{k,r}\sin(\pi(k+1/2)/N) + g'_{k,i}\cos(\pi(k+1/2)/N) \end{aligned}$$

where G_k is a transform coefficient sequence for the first channel;

G'_k is a transform coefficient sequence for the second channel;

$g_{k,r}$ and $g_{k,i}$ are the real and imaginary transform coefficient components of G_k;

$g'_{k,r}$ and $g'_{k,i}$ are the real and imaginary transform coefficient components of G'_k;

Z^{*}_{N-k-1} is the complex conjugate of Z_{N-k-1}; and

$\gamma(k) = \pi(2k+1)/4$.

The modified discrete cosine transform equation can be expressed as

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(2\pi*(2n+1)*(2k+1)/4N + \pi*(2k+1)/4) \qquad k=0...(N/2-1)$$

where *x[n]* is the input sequence for a channel and N is the transform length.

Instead of evaluating X_k in the form given above it could be computed as

$$\begin{aligned} X_k = {} & \cos\gamma*(g_{k,r}\cos(\pi(k+1/2)/N)-g_{k,i}\sin(\pi(k+1/2)/N)) \\ & -\sin\gamma*(g_{k,r}\sin(\pi(k+1/2)/N)+g_{k,i}\cos(\pi(k+1/2)/N)) \\ & g_{k,r}, g_{k,i} \in \Re(\textit{set of real numbers}) \end{aligned}$$

where $G_k = g_{k,r} + jg_{k,i} = \sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$. The symbol j represents the imaginary number $\sqrt{-1}$. The expression $\sum_{n=0}^{n=N-1} (x[n]e^{j\pi n/N}) * e^{j2\pi nk/N}$ is obtained from the well known FFT method, by first using transformation $x'[n]=x[n] * e^{j\pi n/N}$ and then computing the FFT $G_k = \sum_{n=0}^{n=N-1} x'[n] * e^{j2\pi nk/N}$.

For a two channel approach, a complex variable $z[n] = x[n]*e^{j\pi n/N} + jy[n]*e^{j\pi n/N}$ is defined, where $x[n]$ and $y[n]$ are sample sequence for the two channels and $e^{j\pi n/N}$ represents the pre-multiplication factor. Using FFT approach, the frequency coefficient Z_k for the variable $z[n]$ is computed. From Z_k the value $G_k = (Z_k + Z^{*}_{N-k-1})/2$ and $G'_k = (Z_k - Z^{*}_{N-k-1})/2j$, required to compute the final MDCT for each channel, respectively, is calculated.

If either or both the channels require short length transformers, two short transforms are taken using the above approach. If neither need short transform, a single long transform is used. As an additional step in reducing computation, the windowing function can be combined with the pre-processing stage.

Brief Description of the Drawings

The invention is described in detail hereinafter, by way of example only, with reference to preferred embodiments thereof and with aid of the accompanying drawings, wherein:

Figure 1 is a diagrammatic representation of a stream of audio data and the substructure arrangement thereof;

Figure 2 is a functional block diagram of a digital audio encoder;

Figure 3 is a functional block diagram of a system for encoding a single audio channel; and

Figure 4 is a functional block diagram of a system for encoding a pair of audio channels.

Detailed Description of the Preferred Embodiments

The above mentioned Advanced Television Systems Committee (ATSC) Standard document entitled "Digital Audio Compression (AC-3) Standard" (Document A/52, 20 December, 1995) describes methods for encoding and decoding audio signals, and is hereby expressly incorporated herein by reference.

In general, the input to an audio coder comprises a stream of digitised samples of the time domain analog signal. For a multi-channel encoder the stream consists of interleaved samples for each channel. The input stream is sectioned into blocks, each block containing N consecutive samples of each channel (see Fig. 1). Thus within a block the N samples of a channel form a sequence $\{x[0], x[1], x[2], \ldots, x[N-1]\}$.

The time domain samples are next converted to the frequency domain using an analysis filter bank (see Fig. 2). The frequency domain coefficients, thus generated, form a coefficient set which can be identified as $(X_0, X_1, X_2, \ldots, X_{N/2-1})$. Since the signal is real only the first $N/2$ frequency components are considered. Here X_0 is the lowest frequency (DC) component while $X_{N/2-1}$ is the highest frequency component of the signal.

Audio compression essentially entails finding how much of the information in the set (X_0, X_1, X_2, ..., $X_{N/2-1}$) is necessary to reproduce the original analog signal at the decoder with minimal audible distortion.

The coefficient set is normally converted into floating point format, where each coefficient is represented by an exponent and mantissa. The exponent set is usually transmitted in its original form. However, the mantissa is truncated to a fixed or variable number of decimal places. The value of number of bits for coding a mantissa is usually obtained from a bit allocation algorithm which for advanced psychoacoustic coders may be based on the masking property of the human auditory system. A low number of bits results in high compression ratio because less space is required to transmit the coefficients. However this causes very high quantization error leading to audible distortion. A good distribution of available bits to each mantissa forms the core of the most advanced encoders.

In some encoders such as the AC-3, the frequency domain transformation of signals is performed by the (MDCT) modified discrete cosine transform (Eq. 1).

$$X_k = \sum_{n=0}^{n=N-1} x[n] * \cos(2\pi*(2n+1)*(2k+1)/4N + \pi*(2k+1)/4) \quad k=0...(N/2-1) \quad \text{Eq. 1}$$

If directly implemented in the form given above, the MDCT requires $O(N^2)$ additions and multiplications.

Single Channel FFT

It is possible to reduce the number of required operations significantly if one is able to evaluate Eq. 1 using the well known Fast Fourier Transform method of J.W. Cooley and J.W. Tukey (1960). The general Discrete Fourier Transform (DFT) is given below (Eq. 2). It requires $O(N^2)$ complex additions and multiplications. By using the Fast Fourier Transform method the DFT in Eq. 2 can be computed with $O(N\log 2N)$ operations only.

$$F_k = \sum_{n=0}^{n=N-1} (x[n]*e^{2\pi jnk/N}) \qquad k=0..N-1 \qquad \text{Eq. 2}$$

Here j is the symbol for imaginary number, i.e. $j = \sqrt{-1}$.

Although it may not be immediately apparent how Eq. 1 can be transformed to Eq. 2, a careful analysis shows that this is indeed possible. To simplify Eq. 1, two functions can be defined

$$\alpha(n,k) = 2\pi(2n+1)(2k+1)/4\mathrm{N} \qquad \text{Eq. 3}$$

$$\gamma(k) = \pi(2k+1)/4 \qquad \text{Eq. 4}$$

Then, using these functions, Eq. 1 can be rewritten as

$$X_k = \sum_{n=0}^{n=N-1} x[n]*\cos(\alpha(n,k)+\gamma(k)) \qquad \text{Eq. 5}$$

$$= \sum_{n=0}^{n=N-1} x[n]*(\cos\alpha(n,k)\cos\gamma(k)-\sin\alpha(n,k)\sin\gamma(k)) \qquad \text{Eq. 6}$$

In Eq. 6 the trigonometric equality, $\cos(a+b) = \cos a \cos b - \sin a \sin b$ is used for simplification. Furthermore, since the function γ(k) is not dependant on variable n, it can be brought outside the summation expression to give

$$X_k = \cos\gamma(k)\sum_{n=0}^{n=N-1} x[n]*\cos\alpha(n,k) - \sin\gamma(k)\sum_{n=0}^{n=N-1} x[n]*\sin\alpha(n,k)$$

$$= T_1\cos\gamma(k) - T_2\sin\gamma(k) \qquad \text{Eq. 7}$$

where $T_1 = \sum_{n=0}^{n=N-1} x[n]*\cos\alpha(n,k)$ and $T_2 = \sum_{n=0}^{n=N-1} x[n]*\sin\alpha(n,k)$

The two terms, T_1 and T_2, can now be evaluated separately. Using Euler's identity $e^{i\theta} = \cos\theta + j\sin\theta$, we can express:

$$\cos\alpha(n,k) = (e^{j\alpha(n,k)} + e^{-j\alpha(n,k)})/2$$

and $\sin\alpha(n,k) = (e^{j\alpha(n,k)} - e^{-j\alpha(n,k)})/2j$.

Therefore we can rewrite the term T_1 as

$$T_1 = \sum_{n=0}^{n=N-1} x[n]*(e^{j\alpha}+e^{-j\alpha})/2 = 1/2(\sum_{n=0}^{n=N-1} x[n]*e^{j\alpha} + \sum_{n=0}^{n=N-1} x[n]*e^{-j\alpha}) = 1/2\ (A_1 + A_2) \qquad \text{Eq. 8}$$

where $A_1 = \sum_{n=0}^{n=N-1} x[n]*e^{j\alpha}$ and $A_2 = \sum_{n=0}^{n=N-1} x[n]*e^{-j\alpha}$

Similarly

$$T_2 = \sum_{n=0}^{n=N-1} x[n]*(e^{j\alpha}-e^{-j\alpha})/2 = 1/2j(\sum_{n=0}^{n=N-1} x[n]*e^{j\alpha} - \sum_{n=0}^{n=N-1} x[n]*e^{-j\alpha}) = 1/2j(A_1 - A_2) \qquad \text{Eq. 9}$$

The term A_1 can thus be evaluated from Eq. 8 and Eq. 9

$$\begin{aligned} A_1 &= \sum_{n=0}^{n=N-1} x[n]*e^{j\alpha} \\ &= \sum_{n=0}^{n=N-1} x[n]*e^{j(2\pi(2n+1)(2k+1)/4N)} \\ &= e^{j\pi(k+1/2)/N} * \sum_{n=0}^{n=N-1} (x[n]*e^{j\pi n/N}) * e^{j2\pi nk/N} \end{aligned} \qquad \text{Eq. 10}$$

If a complex variable is defined as:

$$x'[n]=x[n]*e^{j\pi n/N} \qquad \text{Eq. 11}$$

then Eq. 10 is simply:

$$\begin{aligned} A_1 &= e^{j\pi(k+1/2)/N} * \sum_{n=0}^{n=N-1} x'[n]*e^{j2\pi nk/N} \\ &= e^{j\pi(k+1/2)/N} * G_k \end{aligned} \qquad \text{Eq. 12}$$

where $G_k = \sum_{n=0}^{n=N-1} x'[n]*e^{j2\pi nk/N}$

The complex term $G_k = g_{k,r}+g_{k,i}$, where $g_{k,r}$ and $g_{k,i} \in \Re$ (set of real numbers) in Eq. 12 is essentially the same as F_k in Eq. 2. Therefore the FFT approach can be used to evaluate G_k. This brings down computation from $O(N^2)$ to $O(N\log N)$. Similarly, the second term A_2 in Eq. 8 and Eq. 9 can be evaluated

$$\begin{aligned} A_2 &= \sum_{n=0}^{n=N-1} x[n]*e^{-j\alpha(n,k)} = e^{-j\pi(2k+1/2)/N} * \sum_{n=0}^{n=N-1} (x[n]*e^{-j\pi n/N})*e^{-j2\pi nk/N} \\ &= e^{-j\pi(2k+1/2)/N} * G_k^* \end{aligned} \qquad \text{Eq. 13}$$

where $G_k^* = \sum_{n=0}^{n=N-1} (x[n]*e^{-j\pi n/N})*e^{-j2\pi nk/N}$

Note that G_k^* is actually the complex conjugate of G_k which was obtained by Eq. 12. That is, if $G_k = g_{k,r}+g_{k,i}$, where $g_{k,r}$ and $g_{k,i} \in \Re$ as defined earlier, then $G_k^* = g_{k,r} - jg_{k,i}$. Therefore G_k^* in Eq. 13 does not need to be computed again, and the result from Eq. 12 can be re-used. That is, only one FFT needs to be computed for the evaluation of T_1. The result of Eq. 8 to Eq. 13 is thus

$$T_1 = 1/2(e^{j\pi(k+1/2)/N}\, G_k + e^{-j\pi(k+1/2)/N}\, G_k^*) \qquad \text{Eq. 14}$$

Next, the term T_2 can be analysed

$$\begin{aligned} T_2 &= \sum_{n=0}^{n=N-1} x[n]*(e^{j\alpha}-e^{-j\alpha})/2j \\ &= 1/2j(A_1-A_2) \\ &= 1/2j(e^{j\pi(k+1/2)/N}\, G_k - e^{-j\pi(k+1/2)/N}\, G_k^*) \end{aligned} \qquad \text{Eq. 15}$$

Finally, after simplifications of Eq. 7, 14 and 15

$$
\begin{aligned}
X_k &= \cos\gamma(k)\ 1/2(e^{j\pi(k+1/2)/N}\ G_k + e^{-j\pi(k+1/2)/N}\ G_k^*) \\
&\quad - \sin\gamma(k)\ 1/2j(e^{j\pi(k+1/2)/N}\ G_k - e^{-j\pi(k+1/2)/N}\ G_k^*) \\
&= \cos\gamma\ *\ (g_{k,r}\cos(\pi(k+1/2)/N) - g_{k,i}\sin(\pi(k+1/2)/N) \\
&\quad - \sin\gamma\ *\ (g_{k,r}\sin(\pi(k+1/2)/N) + g_{k,i}\cos(\pi(k+1/2)/N) \\
&= \cos\gamma\ *\ T_1\ -\ \sin\gamma\ *\ T_2 \qquad \text{Eq. 16}
\end{aligned}
$$

The term $G_k = g_{k,r} + jg_{k,i}$ is computed in $O(N\log N)$ operation by use of FFT algorithms. The additional operation outlined in Eq. 16 to extract the final X_k is only of order $O(N)$. Therefore the MDCT can now be computed in $O(N\log_2 N)$ time. The operations required to obtain the MDCT are illustrated in Fig. 3.

Combining Two Channels into Single FFT

Suppose the multi-channel encoder is required to process m audio channels. Instead of computing an FFT for each channel as described in the previous section, it is possible to further reduce the computational requirement of the coder by combining two channels and using a single FFT only. In effect, instead of m FFTs only $m/2$ FFTS need to be computed.

If the input sequence are real numbers then it is known that DFT for any two channels can be computed with only one FFT block by considering the input as a complex number. The real part is formed from the sequence for any one channel and the imaginary part is from data of another channel. After the Fourier Transform is computed for the resulting complex variable, the resulting transform for each channel can be easily retrieved.

However, in the present case the input data to the FFT block is actually a complex number (formed by multiplying the real data by complex variable $e^{j\pi n/N}$). In this case, there is no straightforward way of retrieving the frequency transform after having combined two channels. However, using some processing after the FFT one can still compute the DFT of two channel using a single FFT block.

Let $\{x[0],x[1],x[2],\ldots,x[\mathrm{N}-1]\}$ be N input samples of the first channel and $\{y[0],y[1],y[2],\ldots y[\mathrm{N}-1]\}$ be the samples for the second channel. As described above, the frequency coefficients $G_k = \sum_{n=0}^{n-N-1} x[n]e^{j\pi n/N} * e^{j2\pi nk/N}$ (Eq. 12 and 13) must be obtained for the first channel; and similarly, for the second channel

$$G'_k = \sum_{n=0}^{n-N-1} y[n]e^{j\pi n/N} * e^{j2\pi nk/N}$$

Defining complex variable $z[n] = x[n]*e^{j\pi n/N} + jy[n]*e^{j\pi n/N}$ Eq. 17

and computing its DFT using the FFT method, yields

$$\begin{aligned} Z_k &= \sum_{n=0}^{n=N-1} z[n]*e^{j2\pi nk/N} \qquad k=0..N-1 \\ &= \sum_{n=0}^{n=N-1} (x[n] + jy[n])e^{j\pi n/N}*e^{j2\pi nk/N} \\ &= \sum_{n=0}^{n=N-1} (x[n] + jy[n])*e^{j2\pi n(k+1/2)/N} \end{aligned} \qquad \text{Eq. 18}$$

Now substituting N-k for k in the above expression,

$$\begin{aligned} Z_{N-k} &= \sum_{n=0}^{n=N-1} (x[n] + jy[n])*e^{j2\pi n(N-k+1/2)/N} \qquad k=0..N-1 \\ &= \sum_{n=0}^{n=N-1} (x[n] + jy[n])*e^{j2\pi n(-k+1/2)/N}*e^{-j2\pi n} \\ &= \sum_{n=0}^{n=N-1} (x[n] + jy[n])*e^{j2\pi n(-k+1/2)/N} \end{aligned} \qquad \text{Eq. 19}$$

Since $e^{j2\pi n} = 1$, $n \in \mathrm{I}$ (the set of integers), the term $e^{j2\pi n}$ vanishes in the above expression. Taking the complex conjugate of Z_{N-k}:

$$\begin{aligned} Z^*_{N-k} &= \sum_{n=0}^{n=N-1} (x[n]-jy[n])*e^{-j2\pi n(-k+1/2)/N} \\ &= \sum_{n=0}^{n=N-1} (x[n]-jy[n])*e^{j2\pi n(k-1/2)/N} \end{aligned} \qquad \text{Eq. 20}$$

Using Eq. 18 and 20, separate expressions for G_k and G'_k are required. In a simple case the conjugates in Eq. 18 and 20 should add and subtract to give the required expressions. However in this instance that is not the case. But, substituting *N-k* by *N-k-1* in Eq. 18, the following is obtained

$$Z^*_{N-k-1} = \sum_{n=0}^{n=N-1} (x[n] - jy[n]) * e^{j2\pi n(k+1/2)/N} \qquad \text{Eq. 21}$$

Now the term $e^{j2\pi n(k+1/2)/N}$ is common in both Eq. 17 and 19, and it is possible to isolate.

$$\begin{aligned} Z_k + Z^*_{N-k-1} &= \sum_{n=0}^{n=N-1} x[n] * e^{j2\pi n(k+1/2)/N} + j \sum_{n=0}^{n=N-1} y[n] * E^{j2\pi n(k+1/2)/N} \\ &+ \left(\sum_{n=0}^{n=N-1} x[n] * e^{j2\pi n(k+1/2)/N} - j \sum_{n=0}^{n=N-1} y[n] * E^{j2\pi n(k+1/2)/N} \right) \\ &= 2 \sum_{n=0}^{n=N-1} (x[n] e^{j\pi n/N}) * e^{j2\pi nk/N} \\ &= 2G_k \end{aligned}$$

Similarly,

$$\begin{aligned} Z_k - Z^*_{N-k-1} &= \sum_{n=0}^{n=N-1} x[n] * e^{j2\pi n(k+1/2)/N} + j \sum_{n=0}^{n=N-1} y[n] * E^{j2\pi n(k+1/2)/N} \\ &- \left(\sum_{n=0}^{n=N-1} x[n] * e^{j2\pi n(k+1/2)/N} - j \sum_{n=0}^{n=N-1} y[n] * E^{j2\pi n(k+1/2)/N} \right) \\ &= 2j \sum_{n=0}^{n=N-1} (y[n] e^{j\pi n/N}) * e^{j2\pi nk/N} \\ &= 2jG'_k \end{aligned}$$

That is

$$G_k = (Z_k + Z^*_{N-k-1})/2 \qquad k = 0..N/2-1 \qquad \text{Eq. 22}$$

and

$$G'_k = (Z_k - Z^*_{N-k-1})/2j \qquad k = 0..N/2-1 \qquad \text{Eq. 23}$$

From the expression from Eq. 22 and 23 into Eq. 16, the MDCT for each channel is obtained. The overall process is illustrated in Fig. 4.

Transform Length Adjustment Technique

The frequency transform length *N* is decided by the encoder based on temporal and spectral resolution requirements. The input signal is usually analysed with a high frequency bandpass filter to detect the presence of transients. This information is used to adjust the block length, restricting quantization noise associated with the transient within a small temporal region about the transient, avoiding temporal masking. Thus, if transient is detected in a channel, two short transform of length *N/2* each are taken. In the absence of transient, a single long transform of length *N* is used, thus providing higher spectral resolution.

From the method described in the previous section for computing MDCT for two channels using a single FFT block, it is evident that the transform length for the two paired channels must be the same. Therefore, pairing for the transformation phase much be such that channels with identical transform length are grouped together.

It is however possible that not all channels can be paired with such convenience. Assume that the total number of channels are an even number (if not, take a single FFT for one channel and the rest form an even group). Suppose out of the *m* channels, *l* need long transform and therefore *m-l* require short transform.

If *l* is an even number, then since the total is even, it follows that *l-m* is also even. In this case, from the *l* channels that need long transform, *l/2* pairs are formed and for each of the *l/2* pairs a single FFT is computed to estimate the MDCT for the original paired channels. Similarly, the *l-m* channels are paired to form (*l-m*)/2 pairs and for the (*l-m*)/2 pairs two short FFTs are computed.

Now consider the case when 1 $l = 2r + 1$ is an odd number. Therefore $m - l = 2s + 1$ is also an odd number. The $2r$ channels requiring long transform are paired together to form r pairs and then $2r$ transforms are computed using r FFTs only. Similarly, for the $2s$ channels s pairs are formed. What remains is one channel requiring long transform and another requiring two short transforms. Both of these channels are paired together and two short FFTs are computed to derive the MDCT.

The rationale for constraining the long transform to two short ones is as follows. A short transform is required for restricting quantization noise associated with the transient within a small temporal region about the transient, avoiding temporal masking. A long transform gives slight better frequency resolution but the error is not much compared to the case when in the presence of transient a long transform is utilised. Forcing a long transform onto a channel in the presence of transient leads to greater distortion in the final produced music. This conjecture was proven true by experimental studies on benchmark music streams.

Combining Windowing with pre-processing

Before the time domain signal $x[n]$ is transformed to the frequency domain, a windowing function is usually applied. Thus, if the sampled signal is $p[n]$ then the sequence that is applied to the frequency transformation block is $x[n] = p[n] * w[n]$, where $w[n]$ is the windowing function. From the previous sections we noted that before the FFT is computed for a block a pre-processing is performed as given in Eq. 11 (reproduced below for convenience). Thus

$$\begin{aligned} x'[n] &= x[n] * e^{j\pi n/N} \\ &= (p[n] * w[n]) * e^{j\pi n/N} \\ &= (p[n]* w[n])* (\cos\pi n/N + j \sin \pi n/N) \\ &= p[n] * ((w[n] * \cos\pi n/N) + j(w[n]*\sin\pi n/N)) \qquad \text{Eq. 24} \end{aligned}$$

From Eq. 24 we note that the windowing function can be combined with the cosine and sine multiplication required in Eq. 11. This brings down the computation even further since the sine and cosine are usually implemented in a real time system as table-lookup. If two tables are constructed as defined below

$$r\cos[n] = w[n] * \cos(\pi n/N)$$
$$r\sin[n] = w[n] * \sin(\pi n/N)$$

then Eq. 11 can be rewritten as

$$x'[n] = (p[n] * r\cos[n]) + j(p[n] * r\sin[n]) \qquad \text{Eq. 25}$$

Although the invention has been described herein primarily in terms of its mathematical derivation and application, and the procedures required for implementation, it will be readily recognised by those skilled in the art that the procedures described can be implemented by means of any desired computational apparatus. For example, the invention may be embodied in computer software operating on general purpose computing equipment, or may be embodied in purpose built circuitry or contained in microcode or the like in an integrated circuit or set of integrated circuits.

The foregoing detailed description of embodiments of the invention has been presented by way of example only, and is not intended to be considered limiting to the invention as defined in the claims appended hereto.

Glossary of Equations:

MDCT

$$\begin{aligned} X_k &= \sum_{n=0}^{n=N=1} x[n] * \cos(2\pi*(2n+1)*(2k+1)/4N + \pi*(2k+1)/4) \quad k=0...(N/2-1) \\ &= \cos\gamma * (g_{k,r}\cos(\pi(k+1/2)/N)-g_{k,i}\sin(\pi(k+1/2)/N) \\ &\quad - \sin\gamma * (g_{k,r}\sin(\pi(k+1/2)/N)+g_{k,i}\cos(\pi(k+1/2)/N) \\ &= T_1\cos\gamma(k) - T_2\sin\gamma(k) \end{aligned}$$

$$\begin{aligned} T_1 &= \sum_{n=0}^{n=N-1} x[n]*\cos\alpha(n,k) \qquad & T_2 &= \sum_{n=0}^{n=N-1} x[n]*\sin\alpha(n,k) \\ &= 1/2(A_1 + A_2) & &= 1/2j(A_1 - A_2) \end{aligned}$$

$$\begin{aligned} A_1 &= \sum_{n=0}^{n=N-1} x[n]*e^{j\alpha} \qquad & A_2 &= \sum_{n=0}^{n=N-1} x[n]*e^{-j\alpha} \\ &= e^{j\pi(k+1/2)/N}*G_k & &= e^{-j\pi(2k+1/2)/N} * G_k^* \end{aligned}$$

$$G_k = \sum_{n=0}^{n=N-1} (x[n]*e^{j\pi n/N})*e^{j2\pi nk/N} \qquad G_k^* = \sum_{n=0}^{n=N-1} (x[n]*e^{-j\pi n/N})*e^{-j2\pi nk/N}$$

$$T_1 = 1/2(e^{j\pi(k+1/2)/N}\ G_k+e^{-j\pi(k+1/2)/N}\ G_k^*)$$

$$T_2 = 1/2j(e^{j\pi(k+1/2)/N}\ G_k-e^{-j\pi(k+1/2)/N}\ G_k^*)$$

$$G_k = (Z_k+Z_{N-k-1}^*)/2 \qquad k=0..N/2-1$$

$$G'_k = (Z_k-Z_{N-k-1}^*)/2j \qquad k=0..N/2-1$$

$$\alpha(n,k) = 2\pi(2n+1)(2k+1)/4\mathrm{N}$$

$$\gamma(k) = \pi(2k+1)/4$$